REMARKS

This application has been carefully considered in connection with the Examiner's Office Action dated March 5, 2008. Reconsideration and allowance are respectfully requested in view of the following.

Summary of Rejections

Claims 1-37 were pending at the time of the Office Action.

Claims 1-10, 12, 13, 15-23, and 31-37 were rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 7,159,224 B2 to Sharma et al. (hereinafter "Sharma") in view of U.S. Publication No. 2002/0032783 A1 to Tuatini (hereinafter "Tuatini").

Claims 11 and 14 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Sharma in view of Tuatini as applied to claim 23 above, and further in view of U.S. Publication No. 2005/0223392 A1 to Cox et al. (hereinafter "Cox").

Claims 24-30 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Sharma in view of Tuatini as applied to claim 23 above, and further in view of U.S. Publication No. 2006/0036448 A1 to Haynie et al (hereinafter "Haynie").

Summary of Response

Claims 1 and 33-37 remain as previously presented.

Claims 2-32 remain as originally submitted.

Remarks and Arguments are provided below.

Summary of Claims Pending

Claims 1-37 are currently pending following this response.

Applicant Initiated Interview

Applicants thank examiner Charles Anya for his time and consideration of the arguments in the telephone interview on April 22, 2008. In the interview arguments were presented that Sharma in view of Tuatini did not teach or suggest limitations including the claimed client library and the non-Java application. Examiner Charles Anya indicated that the applied art would be further considered upon receiving this response.

Response to Rejections under Section 103

According to the United States Supreme Court in *Graham v. John Deere Co. of Kansas City*, an obviousness determination begins with a finding that **"the prior art as a whole in one form or another contains all" of the elements of the claimed invention**. *See Graham v. John Deere Co. of Kansas City*, 383 U.S. 1, 22 (U.S. 1966).

Claim 1:

In the Office Action dated November 15, 2007, claim 1 was rejected under 35 U.S.C. § 103(a) as being unpatentable over Sharma in view of Tuatini.

Independent claim 1 recites:

> 1. A method of accessing an Enterprise Java Bean (EJB) by a non-Java application within a computing environment, comprising:
> > a) **making a call**, by the non-Java application, **to a client library**, wherein the call includes input parameters;
> > b) **invoking a function within the client library to construct an HTTP request** corresponding to the input parameters of the call made from the non-Java application;

    c)   passing the HTTP request from the client library to an EjbServlet;

    d)   invoking a method on an EJB by the EjbServlent based upon the HTTP request;

    e)   returning information from the invoked method from the EJB to the EjbServlet;

    f)   decoding the returned information into an HTTP response string by the EjbServlet;

    g)   **transmitting the HTTP response** from the EjbServlet **to the client library**; and

    h)   **parsing and converting the HTTP response by the client library** into return information compatible with the non-Java application and then passing the return information from the client library to the non-Java application.

Applicants respectfully submit that the art of record does not establish a *prima facie* case of obviousness as to the pending claims because the art of record fails to teach or suggest all of the claim limitations. Specifically, Sharma in view of Tuatini fails to teach or suggest a client library and a non-Java application.

I.   <u>Sharma in view of Tuatini does not teach or suggest a client library.</u>

Claim 1 recites, "making a call, by the non-Java application, to a client library ...; invoking a function within the client library to construct an HTTP request ...; transmitting the HTTP response ... to the client library; and parsing and converting the HTTP response by the client library into return information ... and then passing the return information from the client library to the non-Java application." Therefore, the client library performs all of: receiving a call from a non-Java application, invoking a function of the client library, constructing an HTTP request, receiving an HTTP response, parsing and converting the HTTP response into return information, and finally passing the return information to the non-Java application.

For a better understanding of the following arguments, Applicants respectfully refer to paragraphs 0013-0019 of the specification for a discussion of the claimed client library. Paragraph 0014 of the specification discloses, "In an embodiment, the client library 30 is a linkable library that exposes an API ... to allow the applications to make method calls on an EJB." Paragraph 0014 of the specification further discloses, "The client library 30 may be implemented as a single shared object library file ...." Paragraph 0015 of the specification discloses, "In an embodiment, [t]he client library 30 is ... dynamically linked to that application." One skilled in the art at the time of the invention would recognize that the client library may be a module of code that is linked or otherwise interfaced with the non-Java application. For example, in a Unix environment, the client library may be a .so file. Similarly, in a windows environment, the client library may be a .dll file. As another example, the client library may also be a .h file in the C programming language. One skilled in the art at the time of the invention will recognize that there are other types of libraries known to those of ordinary skill in the art at the time of the invention not specifically addressed in the examples above.

The Office Action does not appear to directly indicate which element(s) of Sharma or Tuatini are being read on the claimed client library. The client library does not appear to have been addressed in rejecting the limitation reciting, "making a call ... to a client library." In rejecting this limitation, the Office Action relied on the client side runtime system 134 (i.e. a Java Virtual Machine or JVM), a client side proxy, a javax.xml.rpc.Serviceinterface (i.e., an application program interface or API), and the dynamic proxy as disclosed in column 7, lines 56-67 and column 8, lines 12-53 of Sharma. Applicants respectfully submit that none of these components disclosed by

Sharma are a client library or even have a call being made to them. The only thing making a call in the cited portions of Sharma is the client 130 that may use the above components to make a call to the server 110 to invoke a feature.

In rejecting the limitation reciting, "invoking a function within the client library to construct an HTTP request," the Office Action relied on the client side runtime system 525 and steps 650 and 670 disclosed in column 23, lines 16-33 and lines 52-58 of Sharma. The cited portion of Sharma is directed to the client side runtime system 525 (i.e., a JVM) mapping a remote method call to a SOAP message (i.e., XML format message for distributed environments) based on a mapping between Java types and XML data types and transmitting the SOAP message as part of an HTTP request. As noted above, the client side runtime system 525 is not the claimed client library.

In rejecting the limitation of, "transmitting the HTTP response ... to the client library," the Office Action stated, "'...serve the content...' Col. 6 Ln. 21 - 26, Container 560 "Steps 750/760" Col. 24 Ln. 33 - 40." In column 6, lines 21-26 Sharma discloses that the content is served to the client 130. Accordingly it appears that the Office Action has changed the interpretation of the claimed client library from the client side runtime system 525 (or element 134) to the client 130 itself. Applicants respectfully submit that the client 130 is not the claimed client library. Also, with regard to the citation of the container 560, this element is at the **server** 530 and supports remote procedure calls (RPC) between the server side runtime system 570 and the service endpoint 555 and is not a library, let alone a **client** library as claimed. Further, with regard to steps 750/760 of Sharma disclose carrying return data in a SOAP message as part of an HTTP response back to the client 510. Again, the client 510 itself is not the claimed client library.

14

In rejecting the limitation of, "parsing and converting the HTTP response by the client library into return information ... and then passing the return information from the client library," the Office Action relied on the disclosure of Tuatini. Specifically, the Office Action stated, "'...appropriate format ...' page 14 paragraph 0119, '...DTD for zero or more response messages...' page 15 paragraph 0127, '...translation...' page 16 paragraph 0138." The disclosure in paragraph 0119 of Tuatini corresponds with Fig. 39 and discloses that a passthru component 3950 of the intranet may exchange messages with clients external to the intranet and may also process the messages so that the message is in an appropriate format. The general disclosure of processing the messages to be in an appropriate format is not disclosure of parsing and converting as those terms would be understood by one of ordinary skill in the art at the time of the invention, let alone disclosure of performing those functions by a client library, as claimed.

The disclosure in paragraph 0127 of Tuatini is directed to DTD files that specify the parameters and the parameter types for XML messages received by the client and exchanged with the shared services. One skilled in the art at the time of the inventions would recognize that DTD files define how to interpret XML messages, but do not perform any parsing of the XML messages or any conversion functions. Further, Applicants note that the DTD files are part of the access interface that the dispatcher 4110 of the messaging component 4101 uses to access the shared services and are not a client library, as claimed.

The disclosure in paragraph 0138 of Tuatini is directed to the handling of response messages by the messaging component 4101 from the shared services back to the client. While the cited paragraph does disclose translation, this refers to the translator 4150 that

is used by the messaging component 4101, not to any client library. Further, the translator 4150 is not disclosed to perform any parsing of an HTTP response. Rather, the translator 4150 is used in the creation of an XML response message that will be transported back to the client via HTTP.

Therefore, Tuatini does not disclose parsing and converting the HTTP response by the client library. As discussed above, Tuatini does not provide any disclosure of parsing an HTTP response. Further, none of the translating functionality performed in Tuatini is disclosed to be performed by the client, let alone by the client library, as claimed.

Accordingly, Sharma in view of Tuatini does not teach or suggest a client library.

II.   Sharma in view of Tuatini does not teach or suggest a non-Java application.

The Office Action relied on disclosure of the client side proxy or the dynamic proxy of Sharma to read on the claimed application. As noted in the arguments of section I in the response to the Office Action filed on September 21, 2007, the client side proxy or the dynamic proxy of Sharma is a Java application. For clarity, column 8, lines 40-53 of Sharma are reproduced below:

> "As previously described, an instance of a stub class may represent a client side proxy or a stub instance for a service endpoint. For **a client side proxy**, client 130 **may utilize a dynamic proxy class** that supports a service endpoint interface dynamically at runtime without requiring code generation of a stub class that implements a specific service endpoint interface. **The creation of a dynamic proxy may be supported by the getPort method defined in the javax.xml.rpc.Service interface.** A serviceEndpointInterface parameter associated with this method may specify the service endpoint interface that is supported by the created dynamic proxy. The dynamic proxy may be used by client 130 to invoke an operation on a target service endpoint defined by server 110."

On page 3, the Office Action concedes, "Sharma is silent with reference to a non-Java application." The Office Action relied on disclosure in paragraph [0120] of Tuatini to provide teaching of a non-Java application. Tuatini discloses a framework for enabling clients to access functions of shared services through a messaging component (Tuatini: Fig. 39; paragraph [0114] and [0118]). Tuatini further discloses in paragraph [0126] that each shared service may share a common XML format for messages or that different services could use different formats. Tuatini discloses in paragraph [0126], "For example, Java-based shared services could use Java objects as messages, and other shared services could use XML messages." Paragraph [0120] of Tuatini, which was relied on by the Office Action, states, "The shared services and clients can also include applications supporting various languages and technologies (e.g., both Enterprise JavaBean (EJB) components and non-Java components)." Based on this disclosure, Tuatini discloses that a system may include an application that supports EJB components and an application that supports non-Java components. Also, based on the above cited disclosure of Tuatini and in consideration of the disclosure of Tuatini as a whole, it is apparent that applications that support non-Java components would be applications that support XML. Accordingly, Tuatini does not teach or suggest a non-Java application as claimed, but rather teaches an application that supports non-Java components. As evidenced by the disclosure of Sharma, such an application may be a Java application. In particular, Sharma discloses a Java application that communicates XML messages to a server to invoke a service. The specification discloses in paragraph 0011, "The non-Java application of the present disclosure can be any computing application implementing a business-related functionality written in a non-Java language that requires access to

external functionality encapsulated in an EJB." Paragraph 0011 further provides some examples of non-Java applications as FORTRAN, Visual Basic, C, Pascal, Basic, and ClearBasic.

Accordingly, Sharma in view of Tuatini does not teach or suggest a non-Java application.

For at least the reasons established above in sections I-II, Applicants respectfully submit that all of the limitations of independent claim 1 are not taught or suggested by Sharma in view of Cox and respectfully requests allowance of this claim.

Claims Depending From Claim 1:

Claims 2-10, 12, 13, 15-23, and 31 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Sharma in view of Tuatini.

Claims 11 and 14 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Sharma in view of Tuatini in further view of Cox.

Claims 24-30 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Sharma in view of Tuatini in further view of Haynie.

Dependent claims 2-31 depend directly or indirectly from independent claim 1 and incorporate all of the limitations thereof. Accordingly, for at least the reasons established in sections I-II above, Applicants respectfully submit that all of the limitations of dependent claims 2-31 are not taught or suggested by Sharma in view of Tuatini and respectfully request allowance of this claim. Applicants respectfully submit that neither Cox or Haynie cure the deficiencies of Sharma in view of Tuatini.

Claim 32:

Claim 32 was rejected under 35 U.S.C. § 103(a) as being unpatentable over Sharma in view of Tuatini.

Independent claim 32 remains as originally filed.

Independent claim 32 provides:

32. A computing system for accessing an EJB by a non-Java application comprising:

a) **a non-Java application** in communication with **a client library**;

b) a means for **calling the client library** from the non-Java application wherein said means for calling the client library is used to establish communication between the non-Java application and the client library;

c) an EjbServlet in communication with the client library wherein **the client library comprises a function** to take input parameter information from the call, embed the information into an HTTP request, and transfer the request to the EjbServlet;

d) a means for **transferring information between the client library and the EjbServlet** wherein said means for transferring it is used to establish communication between the EjbServlet and the client library via an HTTP protocol;

e) the EjbServlet configured to **receive the HTTP request from the client library** and invoke a corresponding method on an EJB; and

f) a remote method interface (RMI) for invoking methods and returning Java objects between the EjbServlet and the EJB.

Applicants respectfully submit that claim 32 is not taught or suggested by Sharma in view of Tuatini because they fail to teach or suggest all of the limitations recited claim 32. Specifically, Sharma in view of Tuatini fails to disclose a client library and a non-Java application.

Independent claim 32 includes limitations substantially similar to the limitations discussed in sections I and II above. For at least the reasons established above in sections I and II, Applicants respectfully submit that all of the limitations of independent

19

claim 32 are not taught or suggested by Sharma in view of Tuatini and respectfully request allowance of this claim.

Claims Depending From Claim 32:

Claims 33-37 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Sharma in view of Tuatini.

Dependent claims 33-37 depend directly or indirectly from independent claim 32 and incorporate all of the limitations thereof. Accordingly, for at least the reasons established in sections I-II above, Applicants respectfully submit that all of the limitations of dependent claims 33-37 are not taught or suggested by Sharma in view of Tuatini and respectfully request allowance of this claim.

CONCLUSION

Applicants respectfully submit that the present application is in condition for allowance for the reasons stated above. If the Examiner has any questions or comments or otherwise feels it would be helpful in expediting the application, he is encourage to telephone the undersigned at (972) 731-2288.

The Commissioner is hereby authorized to charge payment of any further fees associated with any of the foregoing papers submitted herewith, or to credit any overpayment thereof, to Deposit Account No. 21-0765, Sprint.

Respectfully submitted,

Date: _5-27-08_

Rodney B. Carroll
Reg. No. 39,624

CONLEY ROSE, P.C.
5601 Granite Parkway, Suite 750
Plano, Texas 75024
(972) 731-2288
(972) 731-2289 (facsimile)

ATTORNEY FOR APPLICANTS

21